RUMS - Realtime Visualization and Evaluation of Live, Virtual, Constructive Simulation Data

George Soler[*] and Srba Jovic[†]

SAIC/NASA Ames Research Center, Moffett Field, CA, 94035

and

James R. Murphy[‡]
NASA Ames Research Center, Moffett Field, CA, 94035

This paper describes the tools and display capabilities being developed to visualize aircraft flight and trajectory information that is produced during a distributed Live, Virtual, Constructive (LVC) simulation events involving a mix of manned and unmanned aircraft. This paper discusses the design approach and general software architecture of the Remote User Monitor Service (RUMS) application. The simulations and flight tests are conducted over a nationwide network with components communicating via software gateways that translate the messages and provide a location to access and collect data. The data collected is not only archived for post-processing and analysis, but also processed in real-time by RUMS, which allows dynamic viewing of the information via a web browser. This real-time dynamic viewing of the data allows test conductors and researchers to monitor the health of the distributed system and of the data being collected not only in situ where tests are conducted but also at remote locations.

I. Introduction

NASA is conducting simulations and flight tests to determine how to reduce barriers that limit the routine access of Unmanned Aircraft Systems (UAS) into the National Airspace System (NAS). Due to the complexity of these events, they utilize a distributed Live, Virtual, Constructive (LVC) environment that supports mixing of live aircraft into a simulated air traffic control (ATC) airspace[1]. LVC constructs have been used for simulation and flight testing for many years throughout the DoD and have been shown to facilitate live and virtual simulation among distributed partners.[2,3,4] By its nature, a distributed simulation is difficult to manage, as participants may not be co-located. The system is comprised of multiple components, facilities and assets software run over wide area networks and possibly over the Internet. This presents

---

[*] Software Developer, SAIC/NASA Ames Research Center MS 243-6, 94035
[†] Software Manager, SAIC/NASA Ames Research Center MS 243-6, 94035
[‡] Project Engineer, NASA Ames Research Center MS 243-1, AIAA Associate Fellow

American Institute of Aeronautics and Astronautics

not only a challenge in conduct of a simulation or flight-test, but also with the monitoring of the health of the system and the collection of the research data.

A dispersed group of simulation end-users and stakeholders in a distributed simulation environment could therefore benefit from a tool that monitors the simulation's progress from a common data archive. In order to ensure the validity of data being collected, the system needs a mechanism to collect data, and monitor the health of the system. For the UAS in the NAS project, a data collection and monitoring capability has been designed and integrated into the LVC distributed test environment that provides the necessary means to remotely access the system and its evaluation in real-time.

Remote User Monitoring System (RUMS) capability was developed for the NASA Ames Research Center's SimLabs at the Distributed Simulation Research Laboratory (DSRL) for the purpose of providing a standardized, user-friendly, real-time simulation monitoring tool concurrently available to all authenticated end-users. RUMS is built upon Vaadin and Object Fabric technologies that enable monitors of the running simulation visualize key data via a web browser as information is being transmitted through the LVC system. It also provides a single point to capture the data for post processing and analysis.

II. Background

A. UAS in the NAS Project

The desire for a capability to fly government and commercial Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) is of increasing urgency.[5] The application of unmanned aircraft to perform national security, defense, scientific, and emergency management functions are driving the critical need for less restrictive access by UAS to the NAS. UAS represent a new capability that will provide a variety of services in the government (public) and commercial (civil) aviation sectors. The growth of this potential industry has not yet been realized due to the lack of a common understanding of what is required to safely operate UAS in the NAS.

NASA's UAS Integration into the NAS Project is conducting research in the areas of Sense and Avoid/Separation Assurance Interoperability (SAA), Human Systems Integration, Communication, and Certification to support reducing the barriers of UAS access to the NAS.[1] To accomplish this task, the Project will conduct a series of Integrated Human-in-the-Loop (IHITL) and Flight Test activities that integrate key concepts, technologies and/or procedures in a relevant air traffic environment. Each of the integrated events will build on the technical achievements, fidelity and complexity of the previous tests and technical simulations, resulting in a research data that supports the development of regulations governing the access of UAS into the NAS.

B. HLA and LVC Gateway

Figure 1 provides a high-level diagram of an LVC System Configuration that is designed to connect four NASA centers: Ames, Armstrong, Langley and Glenn. NASA Ames provides virtual air traffic control (ATC), constructive aircraft traffic and high fidelity B747 cockpit, a live UAS aircraft and UAS Ground Control Station (GCS) is provided by NASA Armstrong, a virtual UAS simulator with the GCS at NASA Langley, and a UAS surrogate live aircraft flown from NASA Glenn.

2

American Institute of Aeronautics and Astronautics

All participating system components are integrated either through a High Level Architecture (HLA) Toolbox interface, or through the LVC Gateway. HLA is an industry standard that provides a well-defined set of messages, message formats and application programming interface (API) for conducting air traffic simulations and supports the distribution of the test components among the separate test facilities. The version of HLA used by this Project utilizes a version of the IEEE 1516 standard Pitch portable Real Time Infrastructure (RTI) High Level Architecture (HLA) and Federation Object Model (FOM) middleware.[6] The existing HLA distributed environment has the capability to integrate constructive air traffic that is generated by the Multi-Aircraft Control System (MACS), which also contains a robust Air Traffic Control and pseudo pilot capability.[7] In addition, MACS has the capability to publish and subscribe to all required messages (Flight Plan, Flight State, and Trajectory Intent) through an HLA Toolbox. The LVC Gateway will integrate participating components that do not have the capability to connect directly to HLA into the LVC.

HLA is an industry standard that provides a well-defined set of messages, message formats and an API for conducting air traffic simulations and supports the distribution of the test components among the separate test facilities. The virtual UAS simulator is integrated into a representation of a Ground Control Station along with a cockpit display of traffic information (CDTI) used to present SAA conflict alerts and advisories to the pilot. The system components send and receive data to other LVC components through LVC Gateway connected to the HLA network. The constructive aircraft and ATC workstations will be co-located on the same network that communicates with other components via the HLA through their respective HLA Toolboxes. Constructive aircraft provide the required background traffic to create a realistic NAS environment.
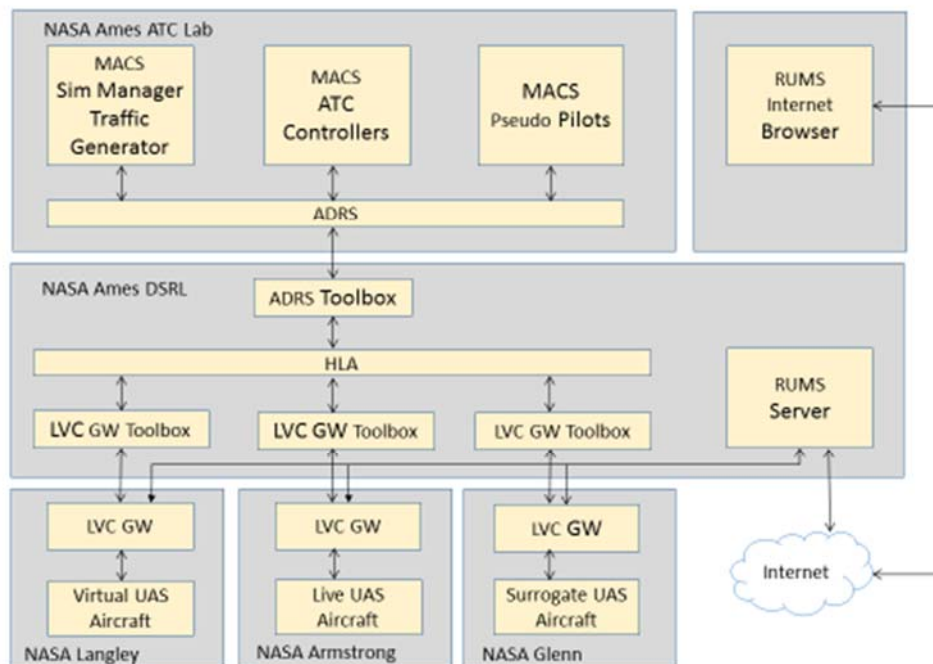
American Institute of Aeronautics and Astronautics

Figure 1. High-level representation of an LVC System Configuration designed to connect four NASA centers.

III.   Remote User Monitoring System (RUMS)

Making simulation data available in real-time to all stakeholders in a distributed environment can be challenging. Presenting real-time simulation data in a simple and universally accessible way usually involves a variety of software tools. Disparate monitoring tools may have features that cannot be easily ported between systems. Some tools may be proprietary or may require some form of maintenance by end-users. All this works against any kind of uniformity in monitoring.

RUMS is an integral component of the Unmanned Aircraft Systems Research Platform (UASRP) subsystem. The UASRP in turn is a part of the DSRL distributed system architecture.
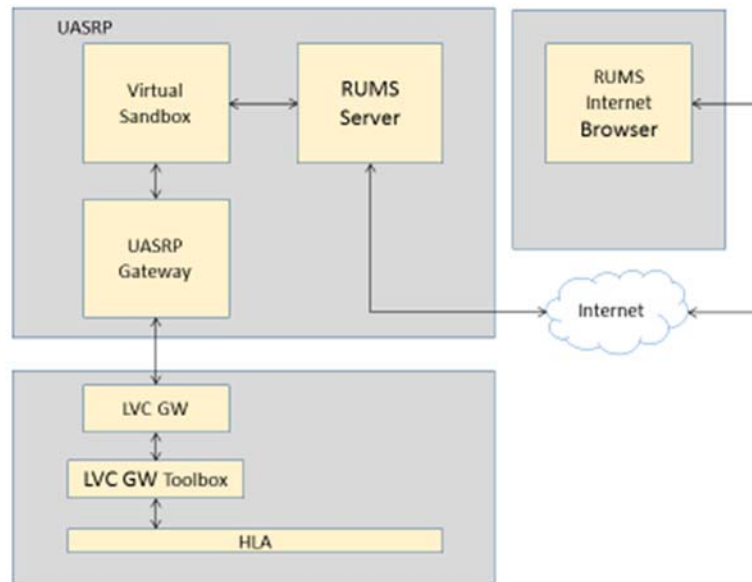

Figure 2. UASRP subsystem architecture diagram and its connectivity to LVC distributed test environment.

For the UI aspects of the design, RUMS uses the Vaadin web development framework. The Vaadin framework interacts transparently with the Google Web Toolkit (GWT). The Vaadin API allows rich UI feature development in pure java with minimal incursions into Javascript. RUMS's main UI features in the web browser include a simulation status page, a traffic display map, a separation display page, and a scenario playback page. Each page has embedded controls and widgets, which taken together allows for a rich user-friendly simulation data viewing experience.

American Institute of Aeronautics and Astronautics

A. Development

RUMS was specifically developed to monitor a distributed air traffic simulation where aircraft state data originates independently in real-time from various networked subsystems. In this context we are referring to either firm or soft real-time data with one-second granularity that is not subject to timing deadlines or hardware interrupts. RUMS was designed to provide airspace situational awareness, to display air traffic control target information, and to provide a simple set of analysis and diagnostics tools for researchers. Although RUMS was developed specifically to assist UAS-NAS integration research, the design, framework selection, and the general architecture can be replicated to fit other problem domains.

Behavior of all UASRP components is controlled by an XML configuration properties file. Each component can be reconfigured, enabled, or disabled between simulation runs by changing the properties file (no re-compilation needed).

The RUMS application is written in java. RUMS is essentially a web application with an embedded web server at its core. RUMS is hosted on a dual-home network system to allow incoming client connections from outside the DSRL environment while protecting the DSRL intranet.

Design of the RUMS communications is predicated on the component-based architecture of the UASRP. RUMS communicates asynchronously with the other UASRP components using a two-tier strategy. Tier-one communications consists of targeted messages between components. These involve low volume network traffic of high importance: sporadic topical command and control messages. Throughput requirements are low but message integrity and persistence is critical and message queuing is needed. Tier-one messaging is handled by a Java Messaging Service (JMS) message-oriented middleware server that runs on a dedicated host. Tier-two communications are based on the transactional memory client-server paradigm. This strategy uses a transactional object model of the aircraft state class to continuously map low-level memory transfers between UASRP networked processes. RUMS uses the ObjectFabric (OF) transactional memory client-server framework. The OF server, which is hosted in its own component, provides a 'Virtual Sandbox' as a means to update and distribute a representation of all the aircraft that participate in the simulation. The OF transactional memory object model description is stored in XML format. As a client of the OF server, RUMS runs a thread that continuously polls the OF server and resolves the OF memory map into multiple message updates that are propagated to the rest of the RUMS client code.

B. Data Monitoring

Data monitoring with RUMS is done through the web interface. At present, there are three main tabs within the browser page that cluster monitoring functions.

American Institute of Aeronautics and Astronautics

The Status tab is the default page. It shows the current DSRL scenario and data run status. It may also show a schedule, contact information, and other general information. The status content changes dynamically: it is updated automatically to reflect current information.

The traffic display tab shows a plan-view map with aircraft markers superimposed on top. The markers are updated automatically in near-time and move at approximately one-second intervals. Each marker is associated with a corresponding aircraft. The aircraft's callsign is shown below the aircraft icon and the altitude is shown below the callsign. A climb (up/green) or descent (down/red) arrow is shown next to the altitude. To view detailed information about a particular aircraft, click on the aircraft marker or select its callsign from the pull-down menu. The traffic display map can be panned and zoomed. Currently, there are two map flavors: the default Remote Monitor Service (RMS) map (stylized with subdued colors), and a standard terrain map.

The separation display tab shows various information panels related to aircraft range, separation, and proximity alerts. The main panel consists of a radar-style display, which shows the targets from an ownship-centric frame of reference.

A playback tab currently under development will add the capability to run pre-recorded scenarios selected by the user.

IV. Conclusion

This paper discusses the design approach and general software architecture of the Remote User Monitor Service (RUMS) application. In addition, the paper describes the tools and display capabilities being developed to visualize aircraft flight and trajectory information that is produced during distributed Live, Virtual, Constructive (LVC) simulations. The simulations and flight tests are conducted over a nationwide network with components communicating via software gateways that translate the messages and provide a location to access and collect data. The data collected is not only archived for post-processing and analysis, but also processed in real-time by RUMS which allows dynamic viewing of the information via a web browser. This real-time dynamic viewing of the data offers test conductors and researchers a common platform to monitor the status of the distributed system.

RUMS is a JAVA web application with an embedded web server at its core that provides a common set of tools and data for analysis and diagnostics used by scientists and researchers in the field of Air Traffic Management. RUMS is a software component in a distributed system of interconnected components that exchange messages. Authenticated remote end-users connect to the RUMS application with a simple web browser without need for add-ons or plugins, making it maintenance-free.

American Institute of Aeronautics and Astronautics

References

[1] NASA Technology Development Project Plan, Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS), 31 January 2013

[2] DoD: "Modeling and Simulation Master Plan", DoD 5000.59P, Oct 1995

[3] Henninger, Amy E., Cutts, Dannie, Loper, Margaret, etal, "Live Virtual Constructive Architecture Roadmap (LVCAR) Final Report", Institute for Defense Analysis, Sept, 2008

[4] Lutz, R., LeSueur, K., Fast, P., Graeff, R., Simolte, A., Rutledge, J., and Motti, R., "A Persistent LVC Simulation Environment for UAS Airspace Integration," The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), December 2010

[5] FAA Modernization and Reform Act of 2012. Feb. 1, 2012. Accessed May 15, 2013 from http://www.gpo.gov/fdsys/pkg/CRPT- 112hrpt381/pdf/CRPT-112hrpt381.pdf

[6] Software website: http://www.pitch.se/prti, May 2014

[7] Prevot, T, "Exploring the many perspectives of distributed air traffic management: The Multi Aircraft Control System: MACS", *International Conference on Human– Computer Interaction in Aeronautics, HCI–Aero 2002*, 23–25.

American Institute of Aeronautics and Astronautics